

# 软件测试—白盒测试

---

## （一）概述

### 基本原理与特点

白盒测试基于软件的源代码，已知产品的内部的工作过程，主要是对程序内部结构进行的测试，白盒测试关注的对象包括两方面：

1. 源代码：逻辑缺陷、内存管理缺陷、数据定义与使用缺陷
2. 程序结构：通过函数调用图，算法流程图等反映程序设计的相关图表，找到程序设计的缺陷

### 两大优势

1. 针对性强，测试效率高，通过不同的白盒覆盖指标有助于衡量对被测对象的覆盖测试程度
2. 在函数级别开展测试工作，缺陷修复的成本低，同时决定了，对测试人员的要求高，技术能力不够的人无法做白盒测试

### 适用阶段

1. 当测试结果为函数时
2. 当测试结果为功能时

### 测试方法的评价

一般可根据覆盖指标的强弱来选择对应的测试方法

## （二）静态白盒测试

### 特点

不需要实际运行被测软件（不需要执行代码，只需使用相应的静态检查软件扫描缺陷），而是直接对软件形式和结构进行分析

### 静态白盒测试的主要内容（重点）

1. 代码检查
2. 静态结构分析
3. 代码质量度量

## （三）静态结构分析

### 基本原理

静态结构分析通过引入多种形式的图表（函数关系调用图，模块控制流图），帮助人们快速了解程序设计和结构，以找到程序设计缺陷与代码优化方向

### 函数关系调用图的测试重点

1. 函数之间的调用关系是否符合要求
2. 是否存在递归调用
3. 函数调用是否层次太深

4. 是否存在孤立的函数

## 原则

1. 根节点是要优先测试的
2. 叶子节点是要优先测试的
3. 接口数量多的节点是要优先测试的

## 函数流程控制图的测试重点

1. 是否存在多出口情况
2. 是否存在孤立的语句
3. 环复杂度是否太大
4. 是否存在非结构化的设计

## 代码质量的度量

ISO9126质量模型是最著名的模型之一，几经修改的最新版本为、ISO/IEC9126:2001,该标准规定了软件产品质量的三个质量模型，即外部质量模型，内部质量模型，和使用中质量模型

## 代码质量度量模型的定义

可构建三层代码质量度量模型，该模型分为三层，从上往下依次是

1. 质量因素：对应质量模型的质量特性
2. 质量标准：对应质量模型的子特性，不同的质量因素由多个质量标准组成
3. 质量度量元：规范软件的行为属性。每个质量标准由多个软件度量元组成

## 静态白盒测试小结

静态白盒测试是白盒测试的重要组成部分，它不需要执行程序，而是通过对比标准与规范，检查程序逻辑，直接定位缺陷，从而加快测试进度，降低测试工作量，省去了动态测试所需的测试用例设计，执行和结果检查的工作，使用效率很高。